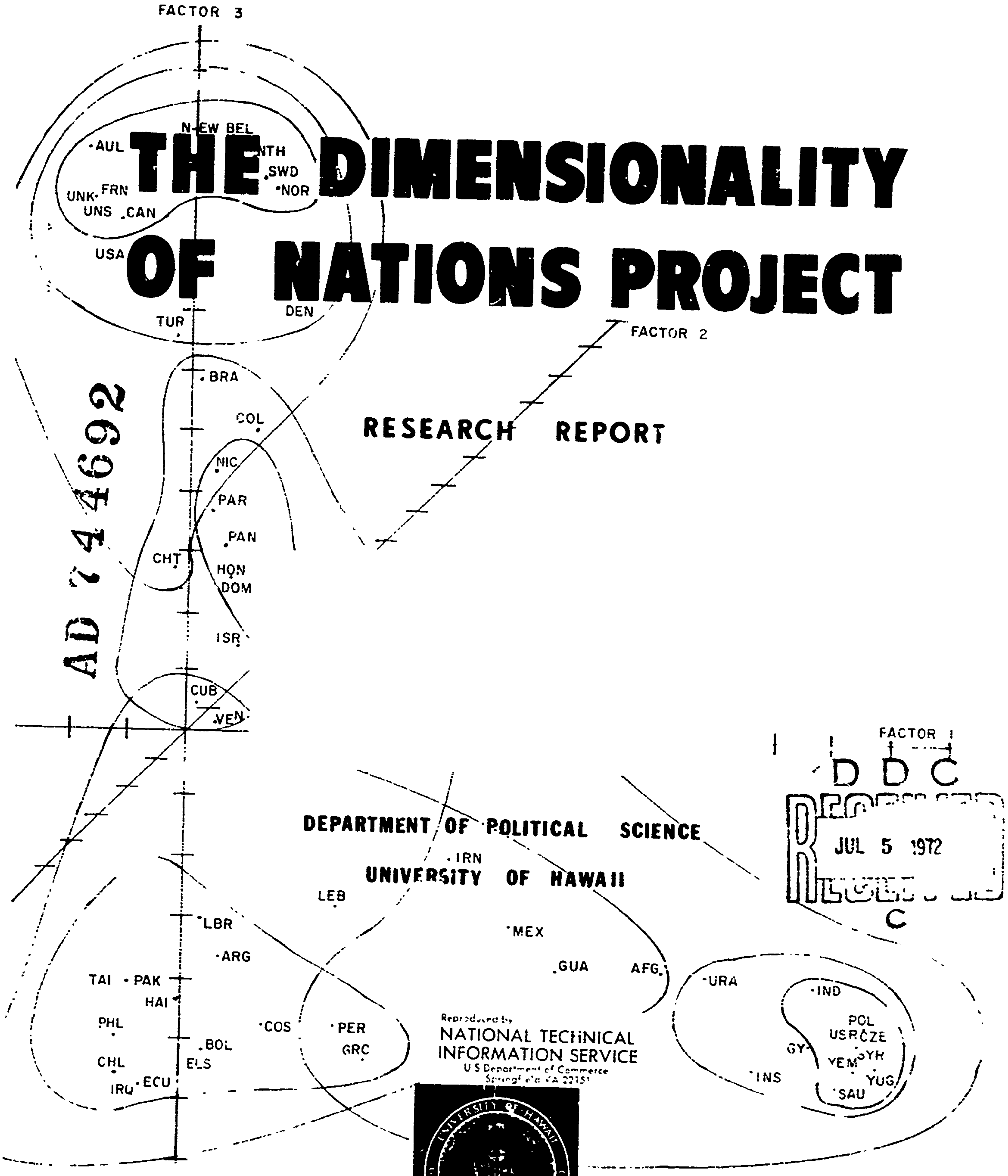
THE DIMENSIONALITY OF NATIONS PROJECT

RESEARCH REPORT



DEPARTMENT OF POLITICAL SCIENCE
UNIVERSITY OF HAWAII

17

THE DIMENSIONALITY OF NATIONS PROJECT
Department of Political Science
University of Hawaii

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 1063

RESEARCH REPORT NO. 62

CODEØ991 PROCEDURE

Charles Wall

May 1972

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-67-A-0387-0003.

The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

This document has been approved for public release and sale; its distribution is unlimited and reproduction in whole or in part is permitted for any purpose of the United States Government.

Ia

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) THE DIMENSIONALITY OF NATIONS PROJECT	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

CODEØ991 PROCEDURE

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
RESEARCH REPORT NO. 62

5. AUTHOR(S) (First name, middle initial, last name)

Charles Wall

6. REPORT DATE MAY 1972	7a. TOTAL NO OF PAGES	7b. NO OF REFS 2
8a. CONTRACT OR GRANT NO N00014-67-A-0387-0003 b. PROJECT NO c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) RESEARCH REPORT NO. 62 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is limited and reproduction in whole or in part is permitted for any purpose of the United States Government.

11. SUPPLEMENTARY NOTES 2500 Campus Road Honolulu, Hawaii 96822	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency Washington, D.C.

13. ABSTRACT

This report describes a Job Control Procedure that allows users with discs to allocate and retrieve data easily. The general methods used to implement the "CODEØ991" (Ø = zero) Procedure may be applied to a wide range of Job Control problems.

This report for the most part is directed to programmers to enable them to implement similar procedures for their own library of programs. It is recommended that non-programmers skip all but the user instructions section.

14

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer						
Computer Programming						
Computer Systems						
Factor Analysis						
Multivariate Analysis						

ABSTRACT

This report describes a Job Control Procedure that allows users with private discs to allocate and retrieve data easily. The general methods used to implement the "CODEØ991" (Ø = zero) Procedure may be applied to a wide range of Job Control problems.

This report for the most part is directed to programmers to enable them to implement similar procedures for their own library of programs. It is recommended that non-programmers skip all but the user instructions section.

TABLE OF CONTENTS

APPENDICES

I. Introduction

The IBM Job Control Language (JCL) is a fairly diverse, relatively complex command language used to allocate and control the Operating System's resources. While programmers using the IBM/360 system need some knowledge of JCL, it is desirable to allow users of production programs to allocate and control the system resources with a minimum of JCL knowledge. In particular, the automatic allocation of Data Definition (DD) statements would facilitate and enhance the use of disc storage for user's data sets.

The "CODEØ991" procedure described in this report was designed to enable users of the DON Project's program library to easily store and retrieve data from the project disc "PEACE1."

The procedure to be described may easily be modified to be used in conjunction with other user disc libraries.

2. Design Objectives

The desirable design objectives for the CODEØ991 procedure are as follows:

- Automatic linkage to DON's program library
- Automatic allocation of all scratch devices
- Automatic allocation of passed data sets for the Modular Factor Analysis package
- Automatic DD allocation of input/output data sets (on disc) requiring only the specification of the data set name.

The automatic linkage of programs via the JCL "STEPLIB" facility and the allocation of scratch devices are standard techniques and will not be discussed further. The automatic allocation (e.g. passing of data sets) for the Modular Factor Analysis package is accomplished by pre-allocation of prescribed data sets for communication between the individual programs in the package. This facility is useful when the user does not want to save the results when the job is finished or desires only to save selected results.

3. Implementation of DD Allocation

The primary purpose of the "CODEØ991" procedure is to allow the automatic DD allocation. This is accomplished by allocating in the procedure nine input (DISP=(NEW,KEEP)) and nine output (DISP=(OLD,KEEP)) devices and setting the DSNAME equal to a symbolic parameter (e.g. DSNAME=&IN31 or DSNAME=&OUT41). Then by setting the default names for each and every DSNAME to NULLFILE (e.g. IN41-NULLFILE and OUT41=NULLFILE) on the PROC DEFAULT statement, allocation will occur only when the user overrides this default and supplies his data set name on the EXEC card.

For example, to run a program named MCORRE and input one data set from the disc while outputing two data sets, the JCL would be as follows:

```
//name JOB (parameters) 'users name'
// EXEC CODEØ991,PROGRAM='MCORRE',
// IN31='MY.INPUT.DATA',
// OUT41='MY.CORREL',
// OUT42='MY.RAW.DATA.BINARY'
```

The user would then allocate in the problem card for the program the devices associated with IN31, OUT41, OUT42. The FORTRAN devices associated with each name are simply the last two digits in the name. The devices a user may allocate in "CODEØ991" are as follows:

INPUT		OUTPUT	
NAME	DEVICE	NAME	DEVICE
IN31	31	OUT41	41
IN32	32	OUT42	42
IN33	33	OUT43	43
IN34	34	OUT44	44
IN35	35	OUT45	45
IN36	36	OUT46	46
IN37	37	OUT47	47
IN38	38	OUT48	48
IN39	39	OUT49	49

Note that in the example, we let the user allocate data sets with qualified names. These are not catalogued data sets, but simply a facility to allow the user to group related data in a tree structure convenient to suit his own particular purposes and data.

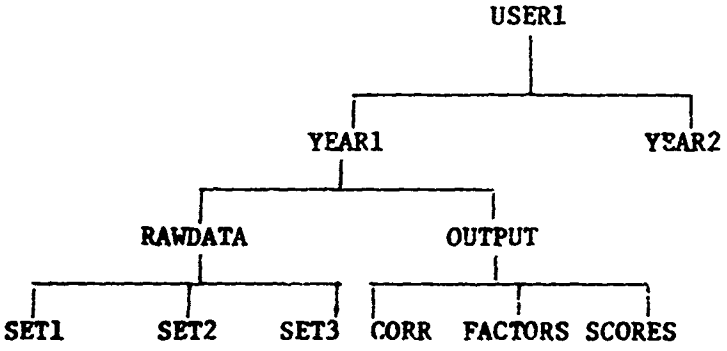
4. Users instructions

The following instructions apply to the "CODEØ991" procedures as it is implemented at the University of Hawaii only. But, only slight modification would be needed for other implementations.

CODEØ991 is a procedure implemented to allow DON Project members easy access to their data sets. The convention adopted is to allow the most general form of data set naming allowed by the system. Hence, users are allowed to form qualified names which may be up to 40 characters in length, and appears in the following format:

'MY.DATA.SET.NAME'

where each name separated by a period "." is no longer than 8 characters and begins with an alpha character. The 40 character length restriction includes the periods. These qualified name structures give rise to a "tree"-like structure which is very useful in organizing data sets for a particular study. One rule prevails, the first name must be the users name. For example,



This diagram gives a part· picture of a typical design of a data library for USER1. To access a particular data set, the user simply codes (starting at the top) the name at each level separated by a period "." to form the qualified data set name. For example,

```
'USER1.YEAR1.RAWDATA.SET2'
'USER1.YEAR1.OUTPUT.CORR'
```

In this fashion, the user can create his own structure to suit his particular needs.

With this general naming convention in mind, then, the user must supply the following information,

- The name of the program to be run
- The region size
- The names of the data sets to be used

To supply the name and region, the user simply codes

```
// EXEC CODEØ991,PROGRAM='name',RG=nnnK
```

where

name = the program name you want to run

nnn = region size in bytes

To access a data set which already exists on the disk (PEACE1), the user codes,

```
// EXEC CODEØ991,PROGRAM='name',RG=nnnK,
// IN32='USER1.YEAR1.RAWDATA.SET1'
```

Notice that a comma must appear after the RG parameter. To create a new data set on PEACE1 the user codes,

```
// EXEC CODEØ991,PROGRAM='name',RG=nnnK,
// OUT43='USER1.YEAR1.OUTPUT.CORR'
```

In particular, if a user desires to run correlation (MCORRE) and determines that the region size is 200K, he would code,

```
//SAMP JOB(time, job#, 200KR), 'user's name'
// EXEC CODEØ991,PROGRAM='MCORRE',RG=200K,
// IN32='USER1.YEAR1.RAW DATA.SET1',
// OUT43='USER1.YEAR2.OUTPUT.CORR'
//GO.SYSIN
(program control cards with the input device = 32, output device = 43)
/*
```

Note that the name used for the output device must not already exist. The comma indicates that another procedure control card follows. Device allocation is automatic and if the job does not complete normally, the output data set will be deleted. Remember in any one job step (e.g., // EXEC) there may be up to nine input (IN31-IN39) nine output (OUT41-OUT49) data sets allocated.

BIBLIOGRAPHY

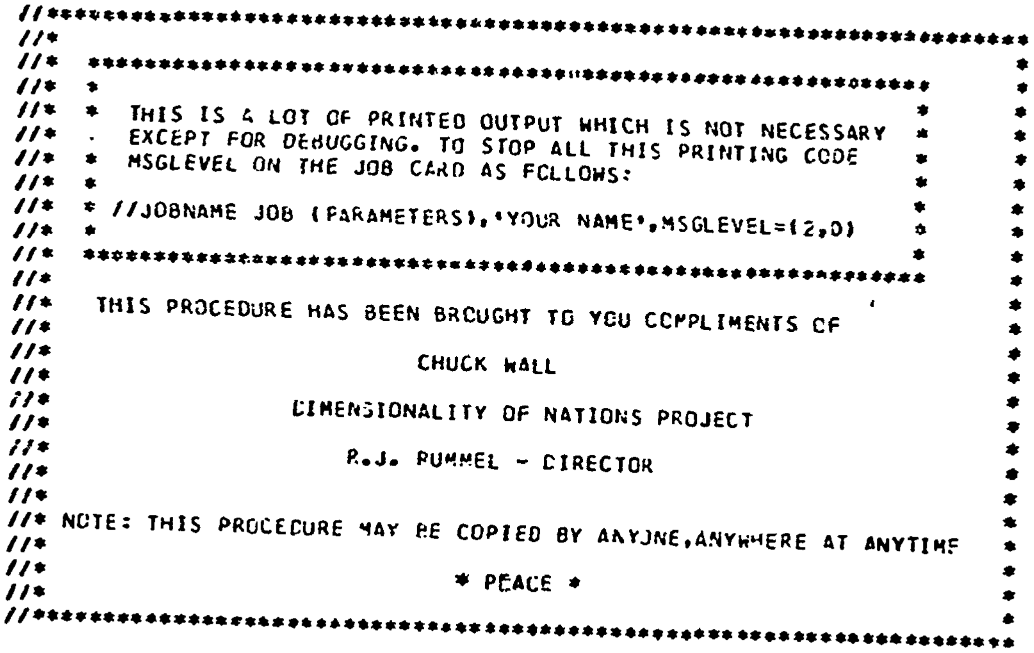
"IBM System/360 Operating System: Job Control Language Reference," GC28-6704.

"IBM System/360 Operating System: Job Control Language Users Guide," GC28-6703.

```
//CODE0991 PROC PROGRAM=DONINFO,RG=200K,PLOT=NULLFILE,
// IN31=NULLFILE,IN32=NULLFILE,IN33=NULLFILE,IN34=NULLFILE,
// IN35=NULLFILE,IN36=NULLFILE,IN37=NULLFILE,IN38=NULLFILE,
// IN39=NULLFILE,
// OUT41=NULLFILE,OUT42=NULLFILE,OUT43=NULLFILE,OUT44=NULLFILE,
// OUT45=NULLFILE,OUT46=NULLFILE,OUT47=NULLFILE,OUT48=NULLFILE,
// OUT49=NULLFILE
//***********************************************************************
//*                                                                     *
//*  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$           *
//*  $                                                      $           *
//*  $   THIS IS A NEW IMPLEMENTATION OF THE CODE0991       $           *
//*  $   PROCEDURE.  THE DATE OF IMPLEMENTATION IS:         $           *
//*  $                                                      $           *
//*  $             ----------------------                   $           *
//*  $             |                    |                   $           *
//*  $             |   20 OCT. 71       |                   $           *
//*  $             |                    |                   $           *
//*  $             ----------------------                   $           *
//*  $                                                      $           *
//*  $   SOME DEVICE ALLOCATIONS HAVE CHANGE, SO CHECK      $           *
//*  $   YOUR DEVICE ALLOCATION TO BE SURE THEY ARE         $           *
//*  $   THEY ARE COMPATIBLE WITH THE NEW DEVICES.          $           *
//*  $                                                      $           *
//*  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$           *
//*                                                                     *
//*    THE CODE PROCEDURE ALLOWS A USER EASY ACCESS TO PROGRAMS         *
//* AND DATA SETS ON PEACE1. THE FOLLOWING IS A LIST OF THE FORTRAN     *
//* UNITS ALLOCATED AND A DESCRIPTION OF THEIR USE.                     *
//*                                                                     *
//*       UNIT      DESCRIPTION                                         *
//*                                                                     *
//*       01        SCRATCH                                             *
//*       02        SCRATCH                                             *
//*       03        SCRATCH - CARD IMAGE                                *
//*       04        SCRATCH                                             *
//*                                                                     *
//*       05        CARD READER                                         *
//*       06        PRINTER                                             *
//*       07        CARD PUNCH                                          *
//*                                                                     *
//*       08        SCRATCH                                             *
//*       09        SCRATCH                                             *
//*       10        SCRATCH                                             *
//*                                                                     *
//*       12         **************************************             *
//*       13         *                                    *             *
//*       14         * MODULAR COMMUNICATIONS DATA SETS   *             *
//*       15         * USED TO PASS DATA SETS FROM ONE    *             *
//*       16         * JOB STEP TO THE NEXT JOB STEP      *             *
//*       17         *                                    *             *
//*       18         **************************************             *
//*                                                                     *
//*       20         **                                                 *
//*       21         * USED BY REVISED DISTANCE ONLY                    *
//*       22         **                                                 *
//*                                                                     *
//*       31             * * * * * * * * * * * * * * * *                *
//*       32            *                               *               *
//*       33           *                                 *              *
//*       34          *                                   *             *
//*       35         *   USER'S INPUT DATA DEVICES         *            *
//*       36          *                                   *             *
//*       37           *                                 *              *
//*       38            *                               *               *
//*       39             * * * * * * * * * * * * * * * *                *
//*                                                                     *
//*       41             * * * * * * * * * * * * * * * *                *
//*       42            *                               *               *
//*       43           *                                 *              *
//*       44          *                                   *             *
//*       45         *   USER'S OUTPUT DATA DEVICES        *            *
//*       46          *                                   *             *
//*       47           *                                 *              *
//*       48            *                               *               *
//*       49             * * * * * * * * * * * * * * * *                *
//*                                                                     *
//***********************************************************************
```

```
//*
//GO EXEC PGM=&PROGRAM,REGION=&RG
//STEPLIB DD UNIT=2314,DISP=SHR,DSNAME=PROGRAMS,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1))
//*
//*     * * * * * * * * * * * *
//*    * ALLOCATE A PLOT TAPE  *
//*     * * * * * * * * * * * *
//*
//PLOTTAPE DD UNIT=(7TRK,,DEFER),LABEL=(,BLP),DSNAME=&PLOT,
// VOL=SER=&PLOT,DISP=(NEW,KEEP)
//*
//*     * * * * * * * * * * * * * * * * 
//*    * ALLOCATE ALL FORTRAN DEVICES  *
//*     * * * * * * * * * * * * * * * *
//*
//FT01F001 DD UNIT=2314,VOL=SER=UHSYS3,DISP=(NEW,DELETE),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(CYL,(2,2))
//FT02F001 DD UNIT=2314,VOL=SER=UHSYS6,DISP=(NEW,DELETE),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(CYL,(2,2))
//FT03F001 DD UNIT=2314,VOL=SER=UHSYS6,DISP=(NEW,DELETE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7280),
// SPACE=(CYL,(2,2))
//FT04F001 DD UNIT=2314,VOL=SER=UHSYS6,DISP=(NEW,DELETE),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(CYL,(2,2))
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=A,DCB=RECFM=UA
//FT07F001 DD SYSOUT=P
//FT08F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(CYL,(2,2))
//FT09F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(CYL,(2,2))
//FT10F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(CYL,(2,2))
//FT12F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=$MODRAW,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT13F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=$MODCOR,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT14F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=$MODFL,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT15F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=$MODRFL,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT16F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=$MODFS,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT17F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=$MODYH,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT18F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=$MODYP,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT20F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
//FT21F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
//FT22F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
```

```
//FT31F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=&IN31,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT32F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=&IN32,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT33F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=&IN33,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT34F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=&IN34,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT35F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=&IN35,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT36F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=&IN36,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT37F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=&IN37,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT38F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=&IN38,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT39F001 DD UNIT=2314,DISP=(OLD,KEEP),DSNAME=&IN39,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294)
//FT41F001 DD UNIT=2314,DISP=(NEW,KEEP,DELETE),DSNAME=&OUT41,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
//FT42F001 DD UNIT=2314,DISP=(NEW,KEEP,DELETE),DSNAME=&OUT42,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
//FT43F001 DD UNIT=2314,DISP=(NEW,KEEP,DELETE),DSNAME=&OUT43,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
//FT44F001 DD UNIT=2314,DISP=(NEW,KEEP,DELETE),DSNAME=&OUT44,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
//FT45F001 DD UNIT=2314,DISP=(NEW,KEEP,DELETE),DSNAME=&OUT45,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
//FT46F001 DD UNIT=2314,DISP=(NEW,KEEP,DELETE),DSNAME=&OUT46,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
//FT47F001 DD UNIT=2314,DISP=(NEW,KEEP,DELETE),DSNAME=&OUT47,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
//FT48F001 DD UNIT=2314,DISP=(NEW,KEEP,DELETE),DSNAME=&OUT48,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
//FT49F001 DD UNIT=2314,DISP=(NEW,KEEP,DELETE),DSNAME=&OUT49,
// VOL=(PRIVATE,RETAIN,SER=(PEACE1)),
// DCB=(RECFM=VBS,LRECL=729,BLKSIZE=7294),
// SPACE=(TRK,(5,5),RLSE)
```



```
//*****************************************************************
//*                                                               *
//*  ******************************************************       *
//*  *                                                    *       *
//*  *  THIS IS A LOT OF PRINTED OUTPUT WHICH IS NOT NECESSARY *  *
//*  .  EXCEPT FOR DEBUGGING. TO STOP ALL THIS PRINTING CODE  *   *
//*  *  MSGLEVEL ON THE JOB CARD AS FOLLOWS:                  *   *
//*  *                                                        *   *
//*  * //JOBNAME JOB (PARAMETERS),'YOUR NAME',MSGLEVEL=(2,0)  *   *
//*  *                                                        *   *
//*  ******************************************************       *
//*                                                               *
//*   THIS PROCEDURE HAS BEEN BROUGHT TO YOU COMPLIMENTS OF       *
//*                                                               *
//*                          CHUCK WALL                           *
//*                                                               *
//*               DIMENSIONALITY OF NATIONS PROJECT               *
//*                                                               *
//*                    R.J. RUMMEL - DIRECTOR                     *
//*                                                               *
//* NOTE: THIS PROCEDURE MAY BE COPIED BY ANYONE,ANYWHERE AT ANYTIME *
//*                                                               *
//*                           * PEACE *                           *
//*****************************************************************
```